PHÂN LOẠI BÀI VIẾT CỦA VOZ.VN

Môn học: Nhập môn khoa học dữ liệu
Bài làm được thực hiện bởi: Nhóm 41

- 1712495: Nguyễn Quang Huy
- 1712858: Nguyễn Ngọc Tú

GIỚI THIỆU

- Bài toán
- Hướng giải quyết

BÀI TOÁN

Phân loại bài viết cho voz.vn để chống trường hợp đặt bài viết sai forum

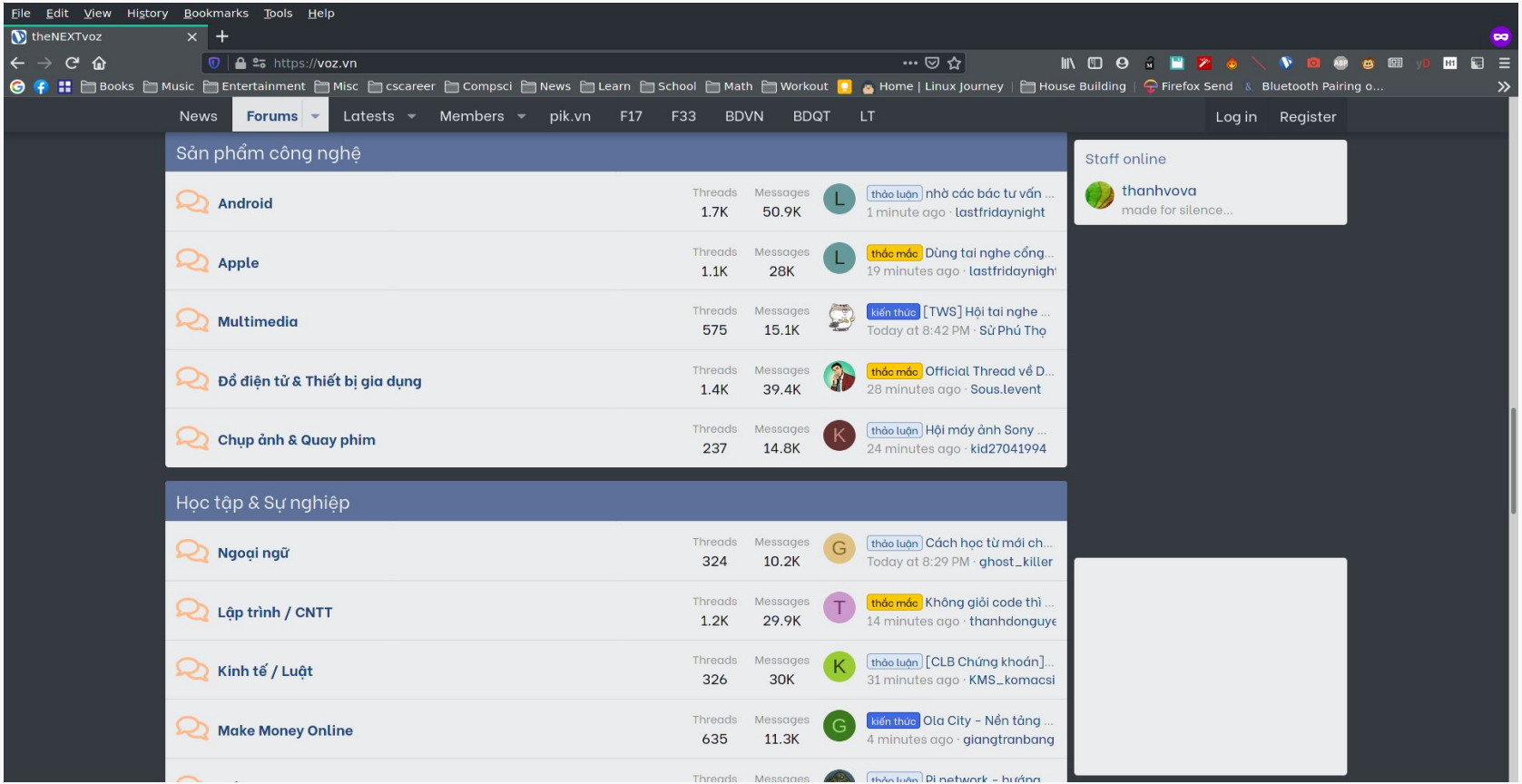
HƯỚNG TIẾP CẬN

- Thu thập dữ liệu: Sử dụng thư viện Scrapy
- Xây mô hình: Sử dụng deep learning Bidirectional LSTM

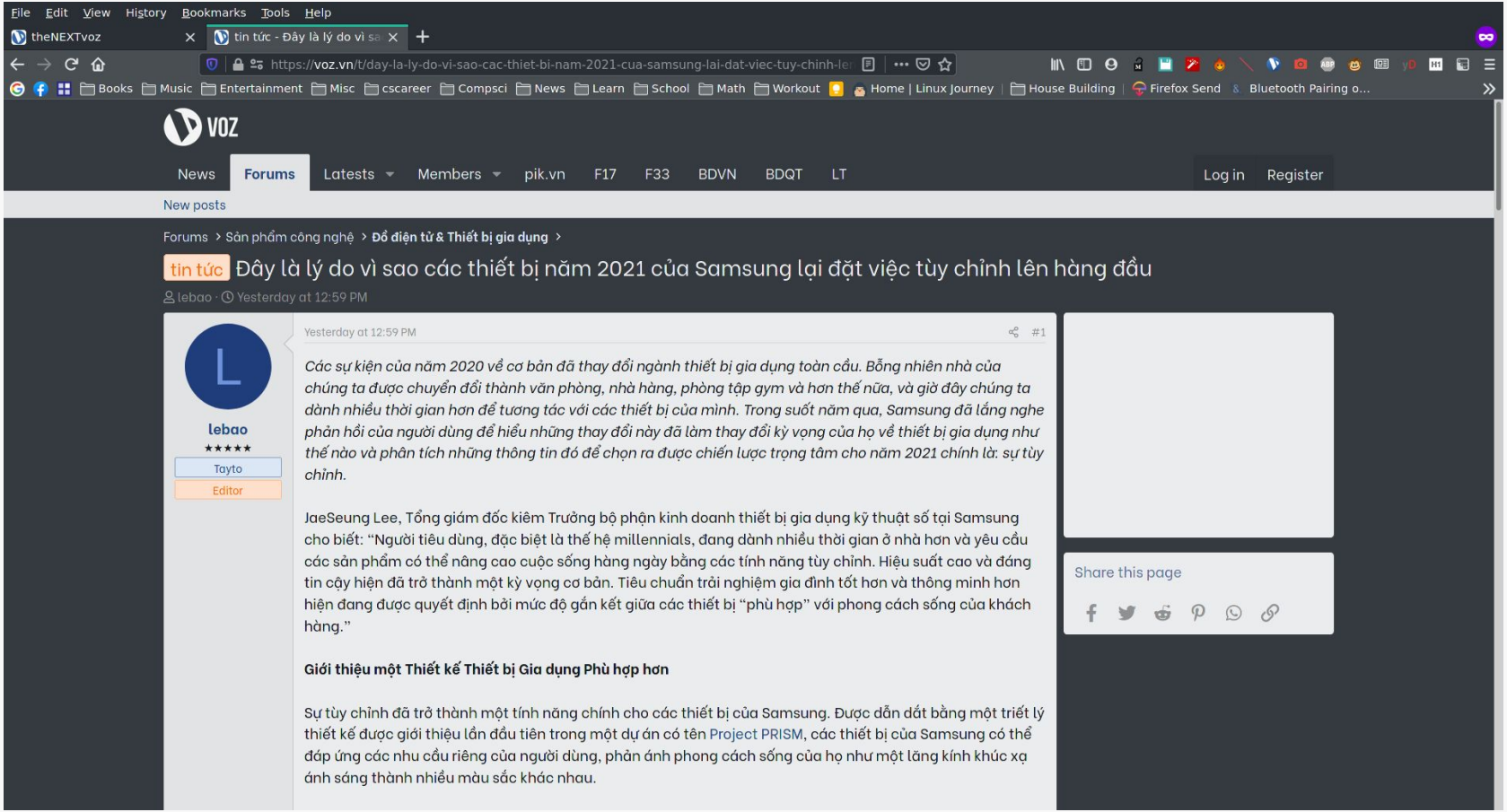
THU THẬP DỮ LIỆU

- Thu thập dữ liệu bài đăng từ website https://voz.vn
- Ở website này, người dùng sẽ tạo bài đăng vào các forum có sẵn như “Điện thoại di động", “Phim / Nhạc / Sách", “Thể dục thể thao”, …
- Ta sẽ lấy về nội dung text trong bài viết của người tạo ra bài đăng, và tên của forum mà bài đăng đó thuộc về

  -> Xây dựng mô hình tự động phân loại bài viết vào forum phù hợp dựa vào nội dung



Trang chủ của voz.vn. Có thể thấy ví dụ một vài forum như **Apple**, **Ngoại Ngữ**,...



Ví dụ một bài đăng thuộc forum **Đồ điện tử & Thiết bị gia dụng,** với tiêu đề: *Đây là lý do vì sao….hàng đầu*

THU THẬP DỮ LIỆU

Dữ liệu được thu thập bằng thư viện Scrapy(https://scrapy.org/) trong Python

Scrapy hỗ trợ việc tạo http request, parse html, lưu kết quả vào file với nhiều định dạng khác nhau. Do đó, với những website có nội dung “tĩnh" như voz.vn thì không cần sử dụng thêm thư viện nào khác.

THU THẬP DỮ LIỆU

Các cột dữ liệu thu thập:

- postTitle: Tiêu đề bài viết
- postContent: Nội dung bài viết
- postLink: URL của bài viết
- forumName: tên forum bài viết thuộc về

Tuy nhiên, trong phần xây dựng mô hình thì chỉ sử dụng postContent và forumName

XÂY DỰNG MÔ HÌNH

Tiền xử lý dữ liệu:

Xóa các dữ liệu có cột nội dung trống

Tách từ:

Do tiếng việt không thể dùng khoảng trắng để phân tách từ, hơn nữa trong một số trường hợp tách từ phức tạp như: (Học sinh học sinh học.) => (, Học sinh, học, sinh học, .,)

=> Cần phải có mô hình tách từ hiệu quả

Sử dụng thư viện ViTokenizer

Link: https://pypi.org/project/pyvi/

Với độ chính xác 0.985

XÂY DỰNG MÔ HÌNH

- Tokenize

  Sử dụng thư viện Tokenizer()

  Để encode đoạn văn cần phân loại sang vector

  Thư viện này sẽ tạo từ điển bằng phương thức fit.

  Sau khi có từ điển thì sẽ chuyển từ sang số tương ứng trong từ điển đó.

  Link: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer

XÂY DỰNG MÔ HÌNH

- Padding vector

  Vector thu được ở trên sẽ có chiều dài khác nhau, ta cần padding để các vector này có cùng độ dài.

  Phương pháp padding:

  - Nếu chưa đủ max len thì ta thêm 0 ở trước.
  - Nếu quá max len thì ta cắt phần sau.

  Thư viện hỗ trợ:

  Link: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/sequence/pad_sequences

XÂY DỰNG MÔ HÌNH

- Word Embedding

  Để mô hình hiệu quả thì mỗi vector đại diện cho từ sẽ gần nhau khi biểu diễn không gian vector nếu từ tương ứng là gần nghĩa.
  ví dụ học sinh, sinh viên thì sẽ gần nhau hơn so với học sinh, cầu thủ nhằm tăng độ chính xác của mô hình.
  Để làm điều đó ta sử dụng pretrained model Word Embedding
  Link:
  https://thiaisotajppub.s3-ap-northeast-1.amazonaws.com/publicfiles/baomoi.window2.vn.model.bin.gz

XÂY DỰNG MÔ HÌNH

- Xây dựng mô hình

  Mô hình sử dụng gồm Embedding layer, Bidirectional LSTM, Dense (activation='softmax' => Đầu ra là xác suất cho từng class)

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 100, 300)          9052500
_________________________________________________________________
bidirectional_3 (Bidirection (None, 128)               186880
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 516
=================================================================
Total params: 9,239,896
Trainable params: 9,239,896
Non-trainable params: 0
```

XÂY DỰNG MÔ HÌNH

- Xây dựng mô hình

  Sau khi sử dụng mô hình trình bày ở trên: Ta thu được độ chính xác là:
  Training set : 99%
  Test set : 90%
  Có vẻ bị over fit
  => Sử dụng phương pháp chống over fit

  Sử dụng hàm mất mát mới **regularized loss function**

XÂY DỰNG MÔ HÌNH

```
rnn_model.add(Dense(4, activation='softmax',kernel_regularizer=regularizers.l1_l2(l1=1e-5, l2=1e-4),
    bias_regularizer=regularizers.l2(1e-4),
    activity_regularizer=regularizers.l2(1e-5)))
```

Độ chính xác sau khi sử dụng chống over fit:

Training set: 99%

Test set : 92%